

Compte Rendu du TP9

Encadré par Akssase

Réalisé par Youssef Oubrik - Ginfo4

December 15, 2024

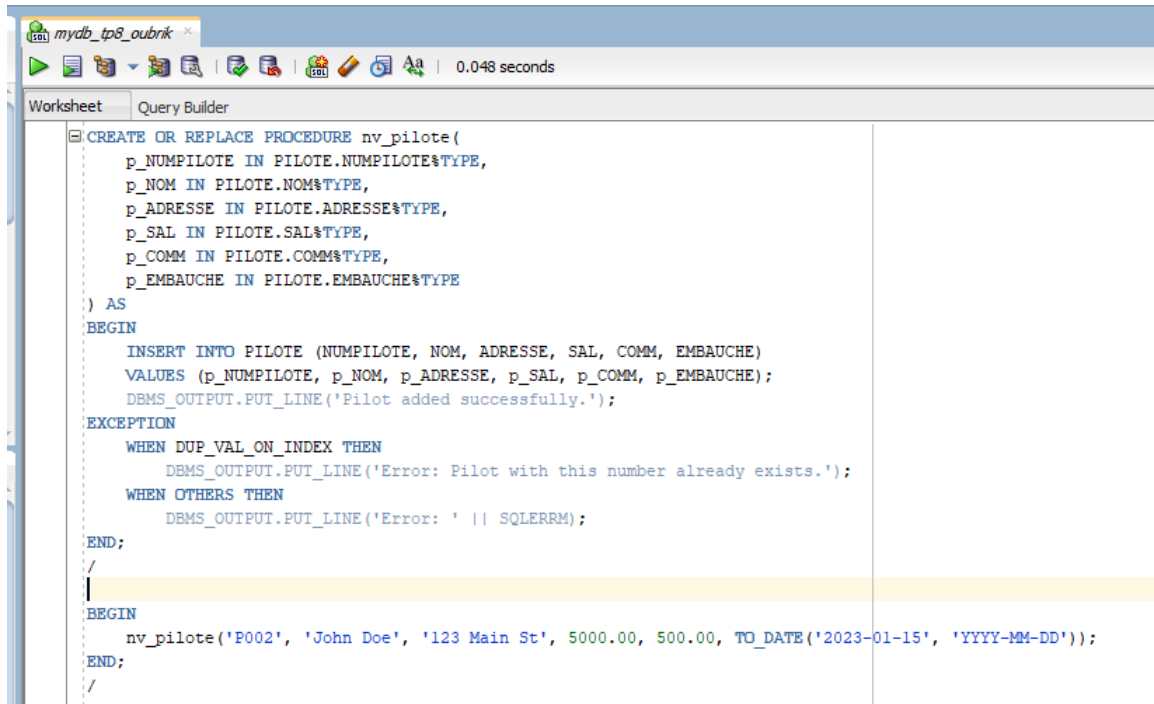
Objectifs du TP9

Le TP9 vise à approfondir les concepts de gestion des bases de données relationnelles avec des exercices pratiques sur :

- Création et gestion de procédures stockées.
- Développement de fonctions stockées.
- Implémentation de déclencheurs.
- Utilisation de packages et transactions.

Procédures Stockées

1. nv_pilote :



The screenshot shows a SQL IDE window titled 'mydb_tp8_oubrik'. The interface includes a toolbar with various icons and a status bar indicating '0.048 seconds'. The main area is divided into two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE OR REPLACE PROCEDURE nv_pilote(  
  p_NUMPILOTE IN PILOTE.NUMPILOTE%TYPE,  
  p_NOM IN PILOTE.NOM%TYPE,  
  p_ADRESSE IN PILOTE.ADRESSE%TYPE,  
  p_SAL IN PILOTE.SAL%TYPE,  
  p_COMM IN PILOTE.COMM%TYPE,  
  p_EMBAUCHE IN PILOTE.EMBAUCHE%TYPE  
) AS  
BEGIN  
  INSERT INTO PILOTE (NUMPILOTE, NOM, ADRESSE, SAL, COMM, EMBAUCHE)  
  VALUES (p_NUMPILOTE, p_NOM, p_ADRESSE, p_SAL, p_COMM, p_EMBAUCHE);  
  DBMS_OUTPUT.PUT_LINE('Pilot added successfully.');
```

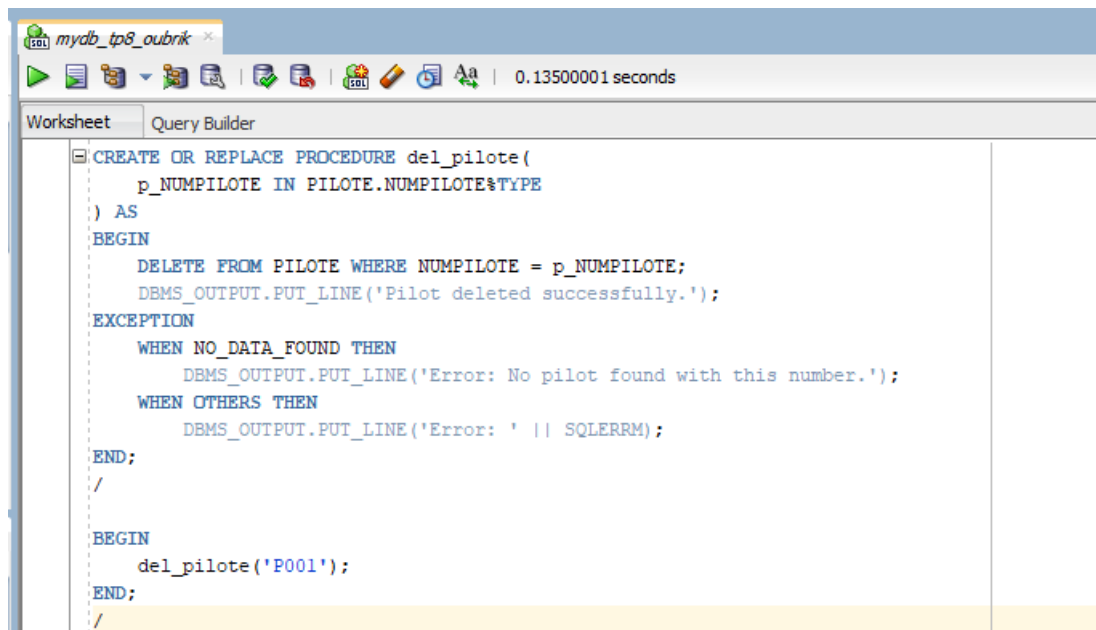
The code is followed by an exception block:

```
EXCEPTION  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE('Error: Pilot with this number already exists.');
```

Then, the procedure is called with specific values:

```
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
END;  
/  
  
BEGIN  
  nv_pilote('P002', 'John Doe', '123 Main St', 5000.00, 500.00, TO_DATE('2023-01-15', 'YYYY-MM-DD'));  
END;  
/  
/
```

2. del_pilote :



The screenshot shows a SQL IDE window titled 'mydb_tp8_oubrik'. The toolbar includes icons for running, saving, and other database operations, along with a timer showing '0.13500001 seconds'. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE OR REPLACE PROCEDURE del_pilote(  
    p_NUMPILOTE IN PILOTE.NUMPILOTE%TYPE  
) AS  
BEGIN  
    DELETE FROM PILOTE WHERE NUMPILOTE = p_NUMPILOTE;  
    DBMS_OUTPUT.PUT_LINE('Pilot deleted successfully.');
```

The code continues with an exception block:

```
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Error: No pilot found with this number.');
```

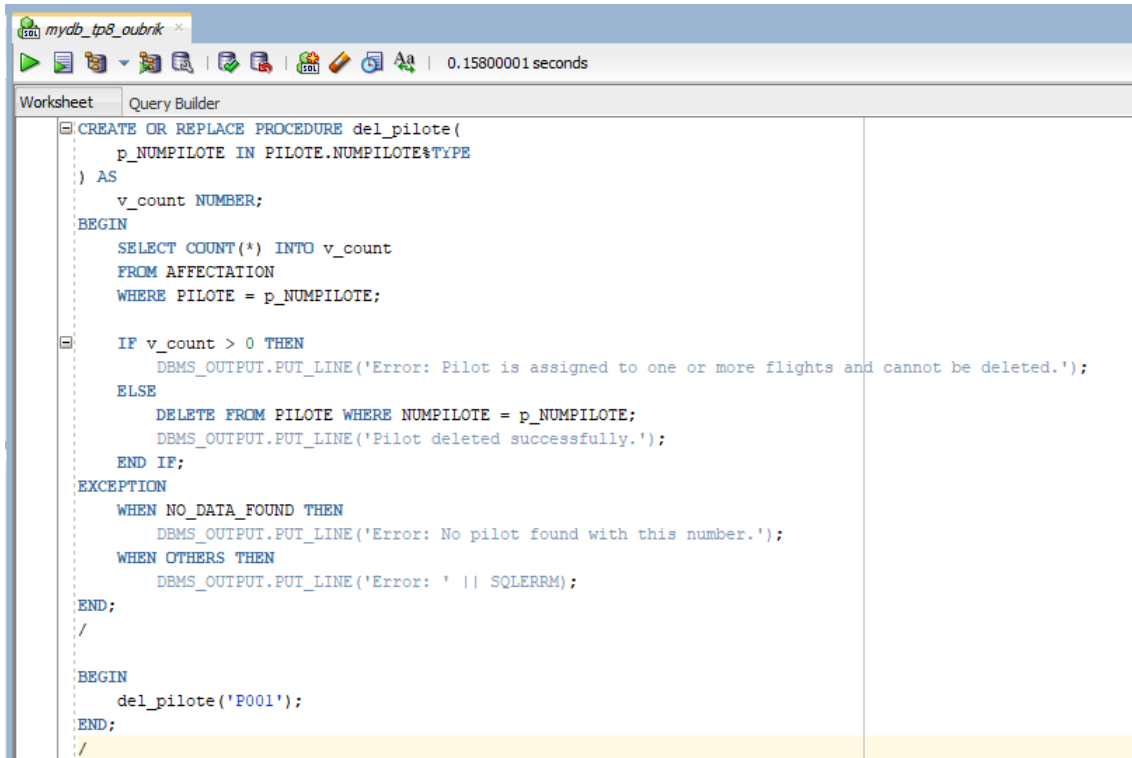
After the exception block, there is a line break, followed by the execution of the procedure:

```
END;  
  
/  
  
BEGIN  
    del_pilote('P001');
```

The code ends with:

```
END;  
  
/
```

3. Modification de del_pilote :



```
CREATE OR REPLACE PROCEDURE del_pilote(
  p_NUMPILOTE IN PILOTE.NUMPILOTE%TYPE
) AS
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count
  FROM AFFECTATION
  WHERE PILOTE = p_NUMPILOTE;

  IF v_count > 0 THEN
    DBMS_OUTPUT.PUT_LINE('Error: Pilot is assigned to one or more flights and cannot be deleted.');
```

```
ELSE
  DELETE FROM PILOTE WHERE NUMPILOTE = p_NUMPILOTE;
  DBMS_OUTPUT.PUT_LINE('Pilot deleted successfully.');
```

```
END IF;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Error: No pilot found with this number.');
```

```
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```

```
/
```

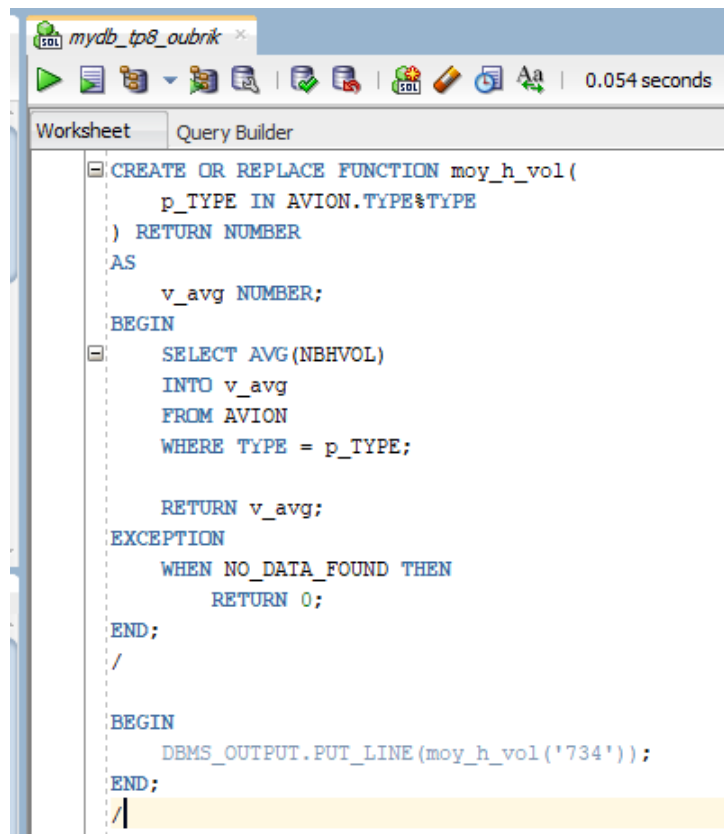
```
BEGIN
  del_pilote('P001');
```

```
END;
```

```
/
```

Fonctions Stockées

4. fonction moy_h_vol :

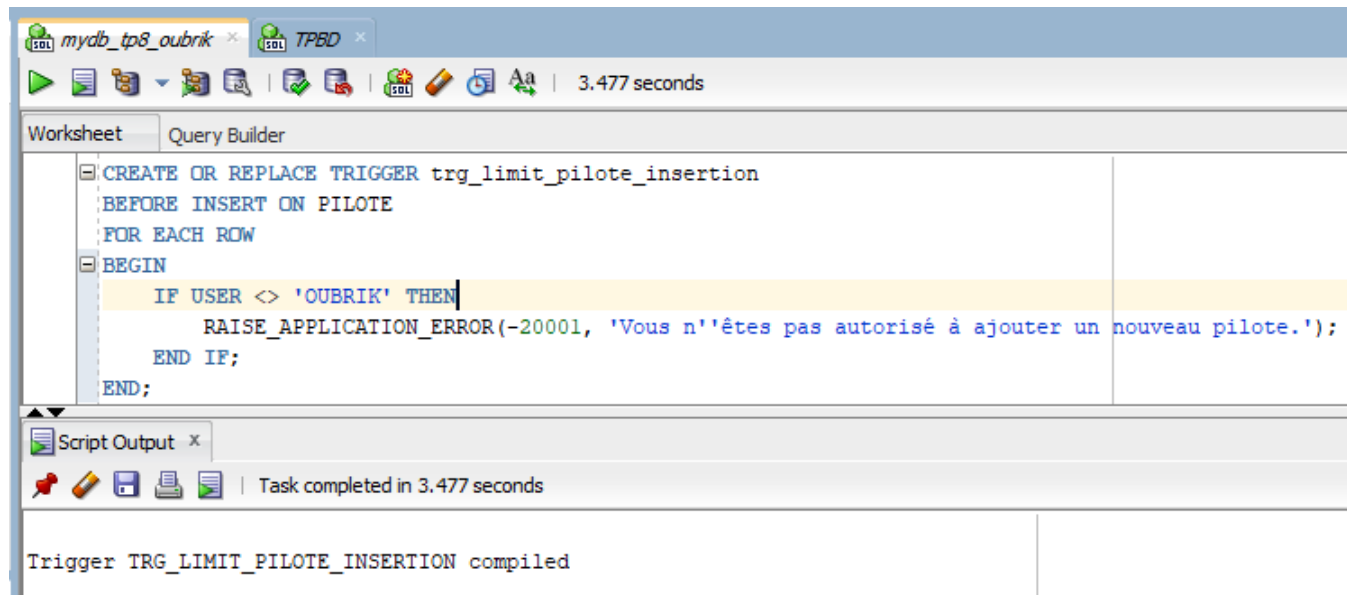


The screenshot shows a SQL IDE window titled 'mydb_tp8_oubrik'. The interface includes a toolbar with various icons and a status bar indicating '0.054 seconds'. The main editor is divided into two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE OR REPLACE FUNCTION moy_h_vol(  
    p_TYPE IN AVION.TYPE%TYPE  
) RETURN NUMBER  
AS  
    v_avg NUMBER;  
BEGIN  
    SELECT AVG(NBHVOL)  
    INTO v_avg  
    FROM AVION  
    WHERE TYPE = p_TYPE;  
  
    RETURN v_avg;  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        RETURN 0;  
END;  
/  
  
BEGIN  
    DBMS_OUTPUT.PUT_LINE(moy_h_vol('734'));  
END;  
/
```

Triggers

5. Limitation de l'ajout d'un nouveau pilote :

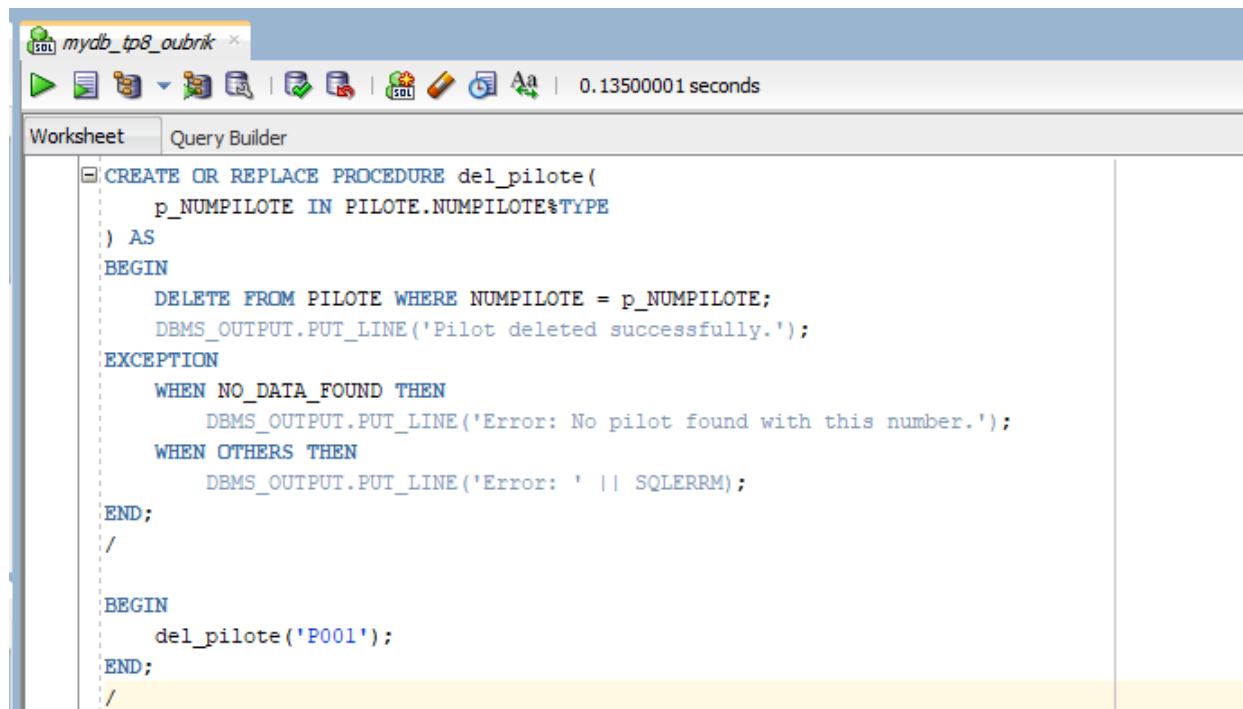


The screenshot shows a SQL IDE window with two tabs: 'mydb_tp8_oubrik' and 'TPBD'. The 'Query Builder' tab is active, displaying a SQL script to create a trigger named 'trg_limit_pilote_insertion'. The script is as follows:

```
CREATE OR REPLACE TRIGGER trg_limit_pilote_insertion
BEFORE INSERT ON PILOTE
FOR EACH ROW
BEGIN
    IF USER <> 'OUBRIK' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Vous n''êtes pas autorisé à ajouter un nouveau pilote.');
```

The script is highlighted in yellow. Below the script, the 'Script Output' tab is visible, showing the message: 'Trigger TRG_LIMIT_PILOTE_INSERTION compiled'. The status bar at the bottom indicates 'Task completed in 3.477 seconds'.

6. trigger verify_nhvol :



The screenshot shows a SQL IDE window titled 'mydb_tp8_oubrik'. The interface includes a toolbar with icons for execution, saving, and editing, and a status bar indicating a duration of 0.13500001 seconds. The main workspace is divided into 'Worksheet' and 'Query Builder' tabs. The 'Worksheet' tab is active, displaying a PL/SQL procedure named 'del_pilote'. The procedure is designed to delete a pilot record from the 'PILOTE' table based on the 'NUMPILOTE' parameter. It includes an exception block to handle cases where no data is found or other errors occur. The procedure is called within a separate block with the value 'P001'.

```
CREATE OR REPLACE PROCEDURE del_pilote(  
    p_NUMPILOTE IN PILOTE.NUMPILOTE%TYPE  
) AS  
BEGIN  
    DELETE FROM PILOTE WHERE NUMPILOTE = p_NUMPILOTE;  
    DBMS_OUTPUT.PUT_LINE('Pilot deleted successfully.');EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Error: No pilot found with this number.');    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
END;  
/  
  
BEGIN  
    del_pilote('P001');END;  
/
```

7. comptabilisation du nombre de lignes modifiées :

The screenshot displays a database query editor interface. The top toolbar includes icons for execution, saving, and undo, along with a timer showing 0.087 seconds. The main window is titled 'Query Builder' and contains the following SQL code:

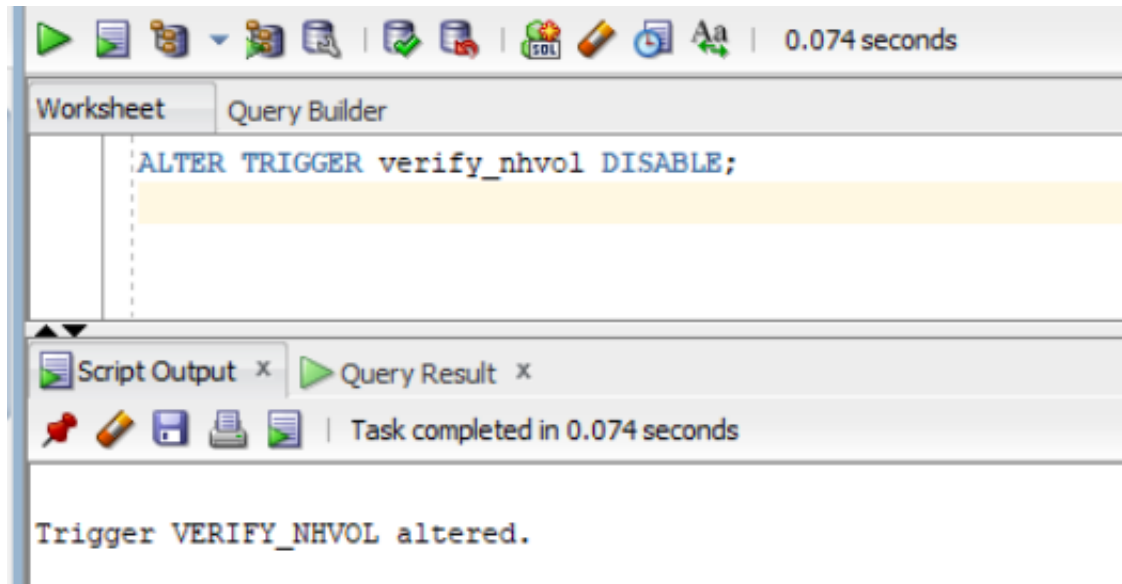
```
CREATE OR REPLACE TRIGGER count_modifications_pilote
AFTER INSERT OR UPDATE OR DELETE ON pilote
FOR EACH ROW
DECLARE
    v_insert_count    NUMBER := 0;
    v_update_count    NUMBER := 0;
    v_delete_count    NUMBER := 0;
BEGIN
    IF INSERTING THEN
        v_insert_count := v_insert_count + 1;
    ELSIF UPDATING THEN
        v_update_count := v_update_count + 1;
    ELSIF DELETING THEN
        v_delete_count := v_delete_count + 1;
    END IF;

    DBMS_OUTPUT.PUT_LINE('INSERT Count: ' || v_insert_count);
    DBMS_OUTPUT.PUT_LINE('UPDATE Count: ' || v_update_count);
    DBMS_OUTPUT.PUT_LINE('DELETE Count: ' || v_delete_count);
END;
```

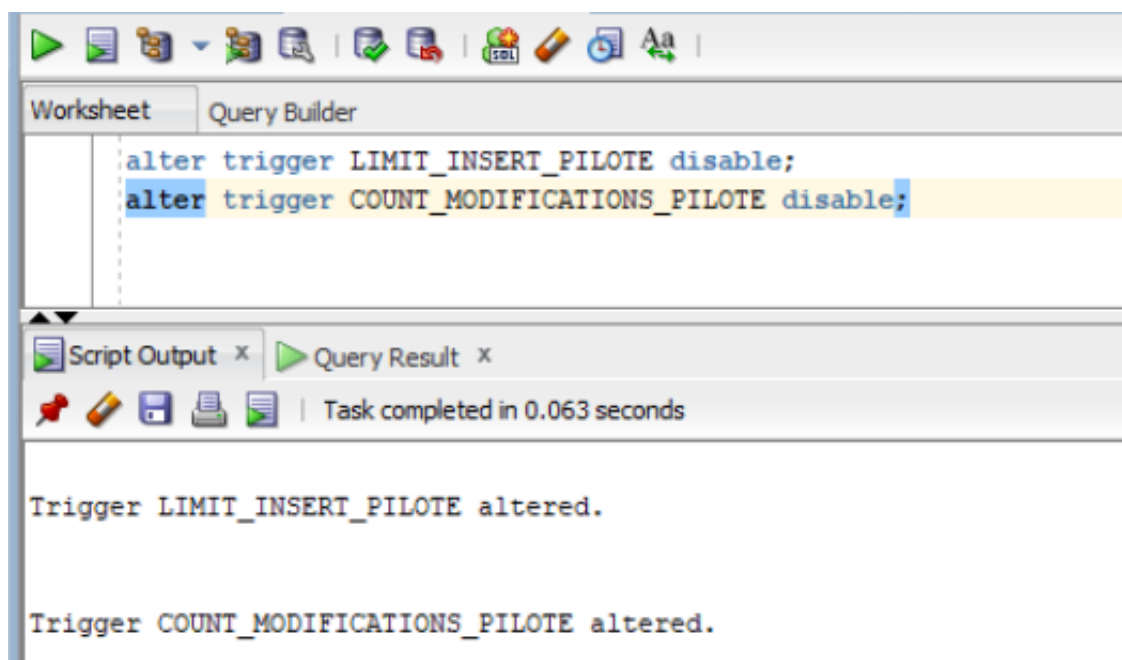
Below the code editor, there is a 'Script Output' tab and a 'Query Result' tab. The 'Script Output' tab is active and shows the message: 'Trigger COUNT_MODIFICATIONS_PILOTE compiled'. The status bar at the bottom indicates 'Task completed in 0.087 seconds'.

8. Désactivation des triggers :

- Désactivation de trigger verify_nhvol.



- Désactivation de tous les triggers de la table PILOTE.



Packages

9. package pour suppression d'un pilote et calcul moyen des heures de vol :

The screenshot displays the SQL Developer interface with the Query Builder tab active. The SQL script being executed is as follows:

```
CREATE OR REPLACE PACKAGE pilote_pkg IS

    PROCEDURE supprimer_pilote(numpilote IN CHAR);

    FUNCTION moyenne_heures_vol(famille IN CHAR) RETURN NUMBER;

END pilote_pkg;
```

The Script Output pane shows the message: "Package PILOTE_PKG compiled". The task was completed in 0.034 seconds.

The second screenshot shows the same SQL Developer interface with a more detailed SQL script. The script includes error handling and database output. The SQL script is as follows:

```
CREATE OR REPLACE PACKAGE BODY pilote_pkg IS

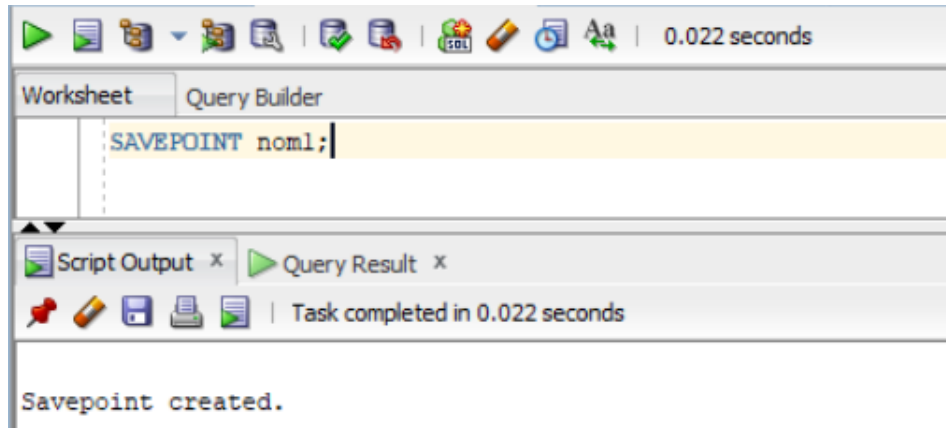
    PROCEDURE supprimer_pilote(numpilote IN CHAR) IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count
        FROM pilote
        WHERE numpilote = numpilote;
        IF v_count = 0 THEN
            RAISE_APPLICATION_ERROR(-20001, 'Le pilote avec le numéro ' || numpilote || ' n''existe pas.');
```

The Script Output pane shows the message: "Package Body PILOTE_PKG compiled". The task was completed in 0.036 seconds.

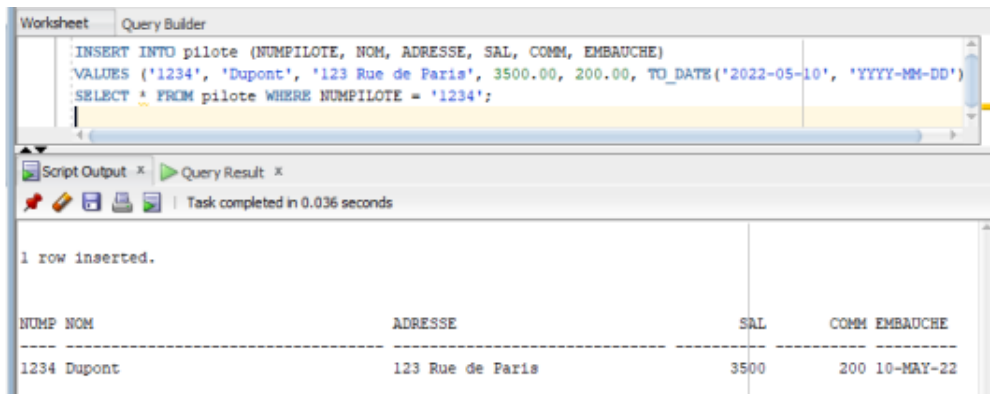
Transactions

10. Visualisation de l'effet d'une transaction :

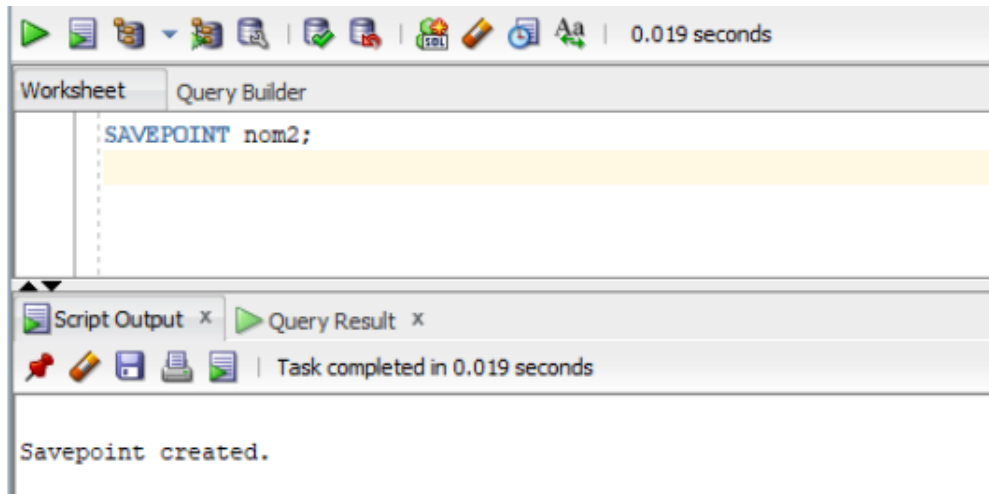
- Création d'un point de sauvegarde : savepoint ;nom1;



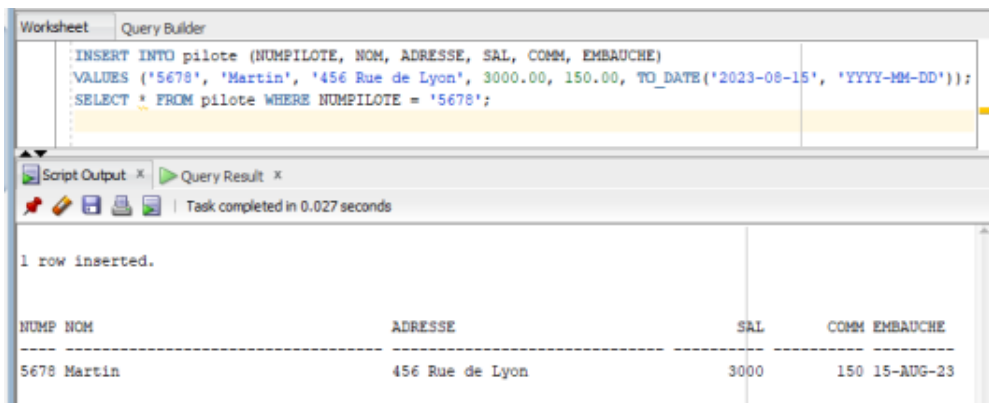
- Insertion d'un pilote, et vérification de son existence. puis validation de la l'insertion.



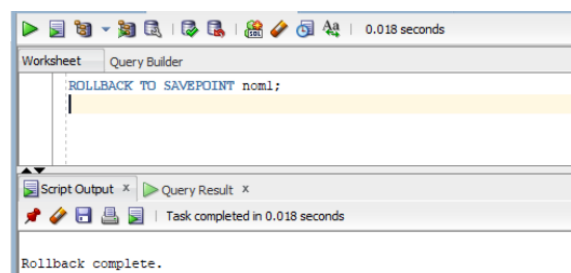
- Création d'un point de sauvegarde : savepoint ;nom2;



- Insertion d'un autre pilote, et vérification de son existence.



- Application de rollback au savepoint jnom1;



- Application de rollback au savepoint jnom2;
- Sélectionner tous les tuples de la table pilote.